access()

Never use simply to avoid changing to a less privileged mode

Sean Barnum, Cigital, Inc. [vita[1]]

2005-10-03

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 7826 bytes

Attack Categories	• Identity Spoofing • Privilege Exploitation
Vulnerability Categories	• Indeterminate File/Path • TOCTOU - Time of Check, Time of Use • Privilege escalation problem
Software Context	• File Management
Location	
Description	The access() function should not be used to attempt to eliminate the need to change to a less privileged mode. The access() function allows one to check the permissions of a file. access() is vulnerable to TOCTOU attacks. It's commonly accepted that one should never use access() as a way of avoiding changing to a less privileged mode. As this is the typical usage, this function should be avoided. On Windows platforms the APIs _access and _waccess are synonymous with access.
APIs	**FunctionName** / **Comments** _access / check _waccess / check access / check
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

	to unintentionally change the state of the targeted resource and yield unexpected and undesired results. The access() call is a check-category call, which when followed by a use-category call can be indicative of a TOCTOU vulnerability. Typically, a user uses access() while in privileged mode to determine whether he would be allowed to access a certain resource in a less privileged mode. If access() returns "true", presumably the program continues and uses that resource. However, in the delay between check and use, the attacker has an opportunity to replace the original resource with something else that might NOT have been allowable.
Exception Criteria	
Solutions	

Solution Applicability	Solution Description	Solution Efficacy
Whenever one is tempted to use access().	On UNIX systems the preferred way of checking for access permissions is to set the EUID/ EGID (i.e. non-root) to the UID/GID of the user running the program before attempting to manipulate the file. Also remember to drop any extra group privileges by calling setgroups(0,0). In this way, the program never makes any attempt at accessing the potentially-restrained resource from a privileged mode.	Effective.
Whenever one is tempted to use access().	Avoid the use of symbolic names, and use	Effective.

		file descriptors when possible.	
	Whenever one is tempted to use access().	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity can not be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.
	Whenever one is tempted to use access().	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Whenever one is tempted to use access().	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
Signature Details	Presence of the access() function call.		

Examples of Incorrect Code	`char filename[]="thefile.txt";`
	`if (0 == access(filename,02))`
	`{`
	`[...]`
	`FILE *theFile = fopen(filename, "w+");`
	`[...]`
	`}`		

Examples of Corrected Code	`char filename[]="thefile.txt";`		

	`if (0 != seteuid(userId)) { /* handle error */ }`
	`if (0 != setegid(userGid)) { /* handle error */ }`
	`if (0 != setgroups(0, 0)) { /* handle error */ }`
	`FILE *theFile = fopen(filename, "w+");`
	`[...]`
	`}`

Source Reference	Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, pp 215-220.
Recommended Resources	• man page for UNIX access() function[2] • MSDN docs for _access() and _waccess()[3]
Discriminant Set	**Operating Systems**: • UNIX • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

1. mailto:copyright@cigital.com